Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación
IIC2233 — Programación Avanzada 2018-2

4 de octubre de 2018

Actividad 08

Excepciones y Testing

1. Introducción

El Departamento de Ciencia de la Computación tras años de éxito en el ~~mercado negro~~ *retail*, al fin decidió abrir un pequeño supermercado en las afueras del campus San Joaquín. La originalidad del DCC ha llevado a bautizar este nuevo negocio como "El DCComercio". Con el tiempo ha desarrollado un sistema para su gestión y una pequeña base de datos con los productos que se venderán. Sin embargo, ajustando el nuevo sistema se dieron cuenta de la gran cantidad de errores que siempre han tenido y han pasado inadvertidos. Esto podría resultar nefasto para la inauguración y por eso te dieron a ti la misión de arreglar la situación.

2. Archivos

En esta actividad deberás trabajar con los siguientes archivos:

- `dccontrolador.py`: Este archivo posee la implementación del programa DCControlador que usará el DCC para gestionar "El DCComercio". Contiene las siguientes clases:
  - `Producto`: Representa a un producto en específico con un nombre, su precio y descuento asociado.
  - `Supermercado`: Representa un supermercado. Cada supermercado posee un nombre y tiene un catálogo de productos. En el catálogo, por motivos de eficiencia, cada producto está asociado a un código único que permite encontrarlo rápidamente.
  - `PedidoOnline`: Esta clase recibe un supermercado y permite agrupar productos para que los clientes puedan comprarlos. Permite que los clientes realicen una orden, que consiste en los productos y la cantidad que desea comprar. Los clientes pueden combinar distintos pedidos, pero deben ser del mismo supermercado.
- `main.py`: Es un pequeño programa que permite importar una base de datos en el sistema DCControlador y efectuar diversos pedidos *online*.
- `datos.csv`: Contiene la base de datos de productos —lamentablemente corrupta— en formato CSV. En cada línea se encuentra el nombre, código, precio y descuento de un producto en particular.
- `tests.py`: En este archivo deberás programar los tests para verificar que el DCControlador esté funcionando correctamente y lo siga haciendo. Los tests a implementar están descritos en la sección 3.2.

3. Instrucciones

3.1. Lanzamiento de errores en el DCControlador

Para evitar que el DCControlador falle silenciosamente en la inauguración, tu primera tarea será hacer visible los errores cuando estos existan. Para esto deberás levantar los siguientes errores en las condiciones que se te indican:

- `ValueError`

  E1 Precio de un producto es seteado con un valor negativo.

  E2 Descuento de un producto no está en el rango de 0 a 0,5.

  E3 Un producto es agregado al supermercado con un código que posee un caracter inválido. Los siguientes caracteres son inválidos: `-&%#@*()`.

  E4 Se intenta agregar una cantidad negativa de un producto al pedido *online*.

- `KeyError`

  E5 Se trata de agregar un producto que no está en el supermercado correspondiente al `PedidoOnline`.

- `RepeatedError`

  E6 Se intenta agregar un producto con un código que ya existe en ese supermercado.

- `InconsistencyError`

  E7 Se intenta sumar dos carros de compra de supermercados distintos.

  Debes utilizar los mensajes de error indicados en la siguiente tabla dependiendo del caso que estás cubriendo:

ID	Tipo de Error	Mensaje del Error
E1	`ValueError`	`'precio base menor que 0'`
E2	`ValueError`	`'descuento no está entre 0% y 50%'`
E3	`ValueError`	`'cantidad menor que 0'`
E4	`ValueError`	`'código posee caracteres inválido: {invalidos}'`
E5	`KeyError`	`'el producto no existe en el supermercado'`
E6	`RepeatedError`	`'{codigo} ya está siendo utilizado'`
E7	`InconsistencyError`	`'carros de compra de distintos supermercados'`

3.2. Testing

Para demostrar que el DCControlador funciona correctamente y seguirá haciéndolo de esta forma en el futuro, el DCC te ha solicitado que programes tests unitarios que verifiquen varias funcionalidades específicas. En el archivo `testing.py`, debes realizar un *test* utilizando `unittest` por cada una de las siguientes condiciones:

- Al intentar ingresar un producto al supermercado con un código que contiene un caracter inválido (`-&%#@*()`) se levanta una excepción del tipo `ValueError`.

- Se puede verificar la existencia o no de un objeto `producto` o un código dentro del catalogo de un supermercado utilizando el *keyword* `in` sobre el supermercado, de la forma: `producto in catalogo`. Este *test* fallará, debes arreglar el código para que pase.
- Un `PedidoOnline` con una orden vacía, tiene un valor total de `$0`.
- Al agregar un producto con un precio de `$x` a un `PedidoOnline`, el valor de este aumente en `$x`.

3.3. Corrección en `main.py`

En esta última parte deberás aprovechar los nuevos errores que arroja el DCControlador para corregir el funcionamiento en `main.py`. Debes lidiar con los errores que te van apareciendo imprimiendo en pantalla el mensaje de error y arreglarlos según la siguiente tabla:

ID	Solución
E1	Asignar precio base de `$1`
E2	Asignar descuento de `0%`
E3	Reemplazar caracter(es) inválido(s) por un `'.'`
E4	El producto no se agrega al pedido
E5	El producto no se agrega al pedido
E6	El producto no se agrega al supermercado
E7	Se imprime un mensaje y no se permite la suma de pedidos.

4. Notas

- Pueden asumir que los únicos errores en la base de datos son los mencionados en el enunciado.
- Recuerda importar tus errores cuando vayas a usarlos.
- Pueden ver documentación de de la clase `Counter` **aquí**.

Entrega

- **Lugar:** En su repositorio privado de GitHub, en la **carpeta** `Actividades/AC08/`
- **Hora del último *push* válido:** 16:00